



Digital Transformation for Systems Engineering

MBSE Symposium

Co-chair: Maged Elaasar (JPL)

Co-chair: Sebastien Gerard (CEA-LIST)

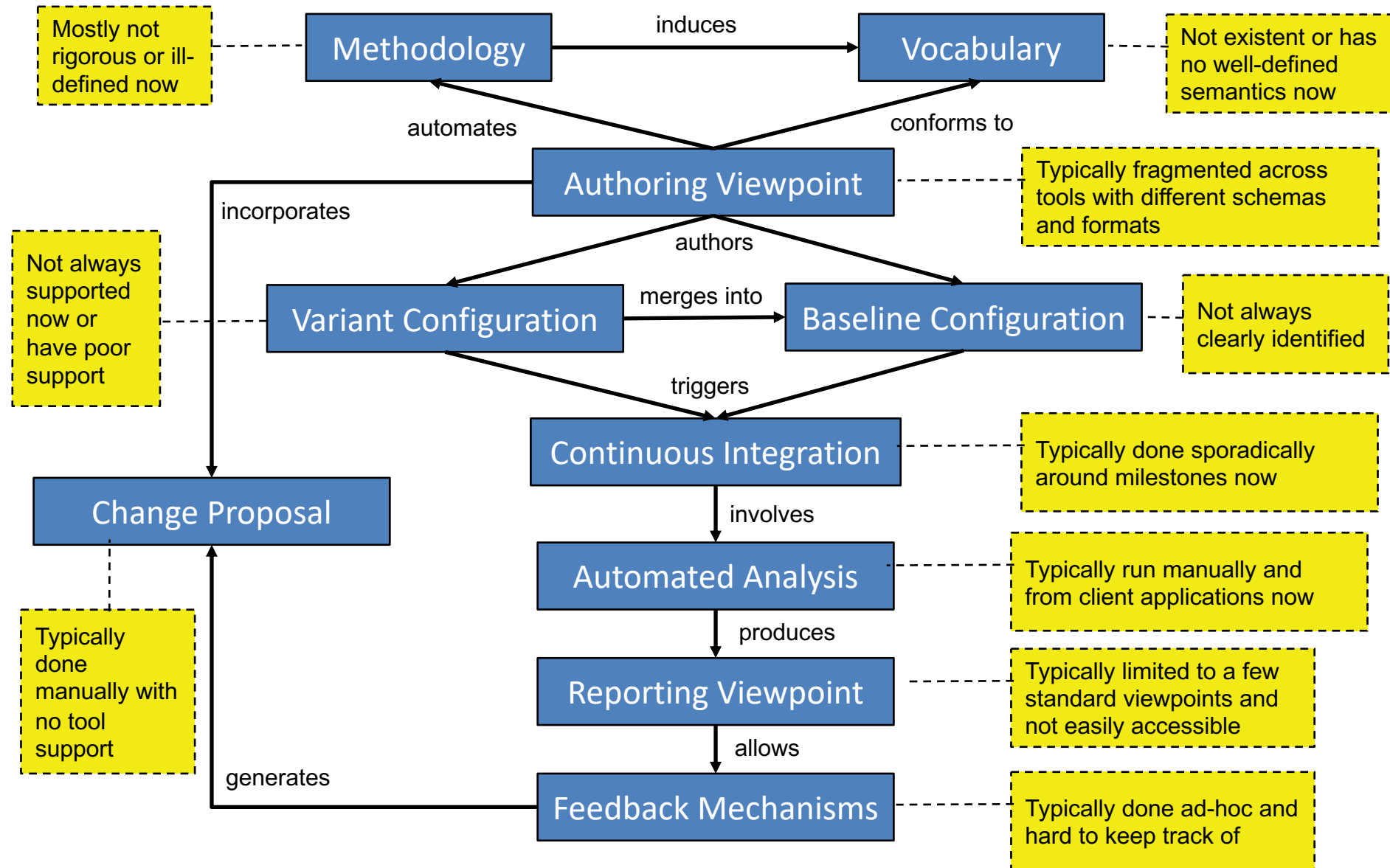
January 24, 2019



Jet Propulsion Laboratory
California Institute of Technology

- Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology

- **Introduction** -5 mins
- **Open CAESAR Initiative**, Maged Elaasar (JPL) -45 mins
- **Papyrus Project**, Sebastien Gerard (CEA-LIST) -10 mins
- **Capella Project**, Stephane Bonnet (Thales) -10 mins
- **Sirius Project**, Stephane Lacrampe (Obeo) -10 mins
- **RangeDB Project**, Harald Eisenmann (Airbus) -10 mins
- **Discussion** -30 mins





INTRODUCING THE

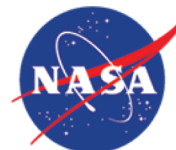
Open CAESAR Initiative

MBSE Symposium

Maged Elaasar, PhD

IMCE Program Chief Architect

January 24, 2019



Jet Propulsion Laboratory
California Institute of Technology

Challenges		Mitigations
System modeling tools are too complex and rigid to use for systems engineers	➔	We need to use domain-specific tools to lower the bar to entry and provide curated libraries and model templates
Different projects create system models differently making it hard to reuse or analyze consistently	➔	We need to define modeling methodologies and facilitate conforming to them using tools
Heterogeneous tools are used in SE because many discipline-specific analyses are needed	➔	We need effective tool integration strategies, facilitating cross tool linking or interchange
Model content is rapidly changing as work progresses, increasing risk, and latency in decision making	➔	We need effective cross-tools configuration management, change management, and continuous automated analysis
Models are being authored by separate teams with different concerns/expertise	➔	We need to enable federated multi-viewpoint architecture with methods to keep the viewpoints aligned
Standard model notation is not sufficient to present their content and communicate it to stakeholders	➔	We need ability to design custom reporting viewpoints , automatically generate and make them accessible

- CAESAR stands for **C**omputer **A**ided **E**ngineering for **S**ystems **A**Rchitecture
 - A software platform to enable an integrated model centric approach to SE
 - Enables a methodology-driven use of SE tools to perform SE functions
 - Supports a federated multi-viewpoint agile systems development process
 - Promotes reuse of existing technology while minimizing vendor lock-in
 - Addresses key challenges of the current practice of MBSE
 - Infuses development best practices learned from software engineering



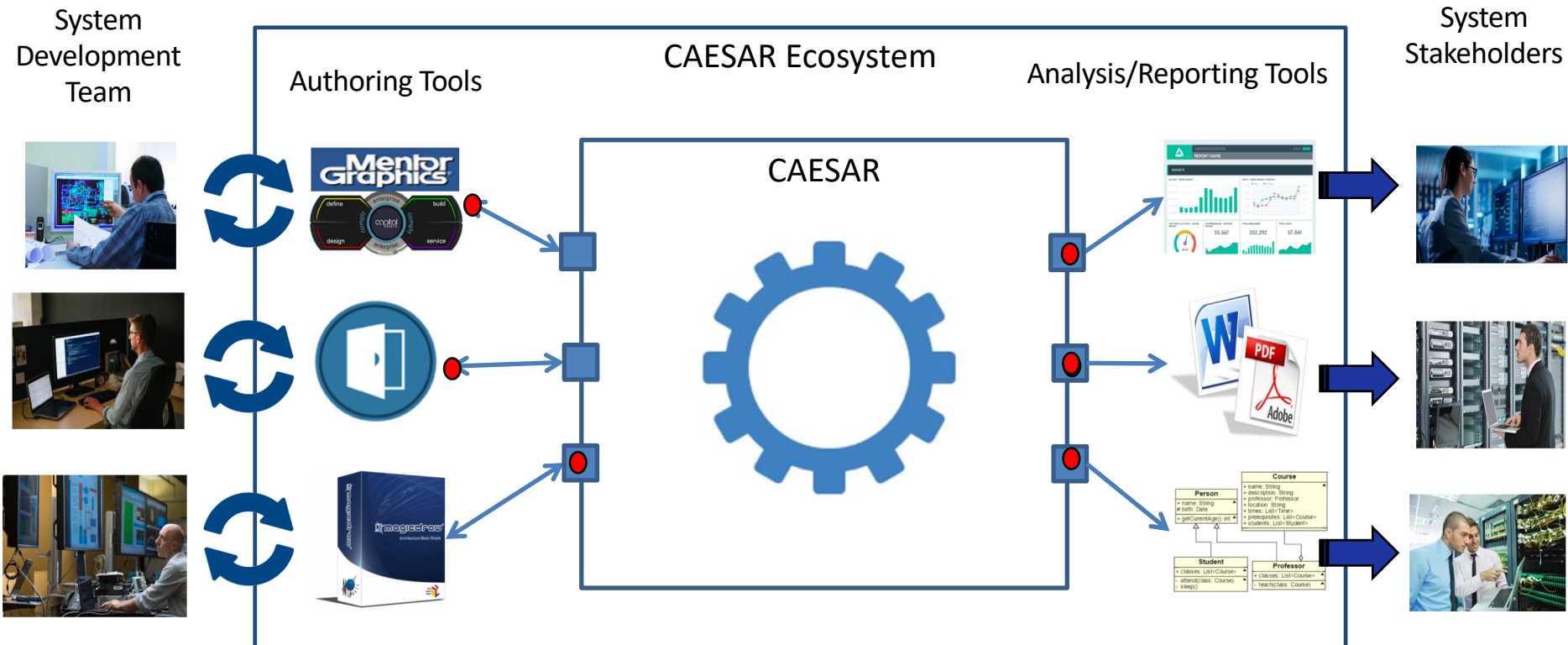
Model Based Systems Engineering
(MBSE)



Integrated Model Centric Engineering
(IMCE)

- **A semantic data warehouse system for MBSE**

- Acquires information from SE authoring tools acting as data sources
- Curates, integrates, configuration manages SE information baseline in one place
- Verifies the consistency of federated information and measures precise differences
- Analyzes information, generates reports and proposes changes



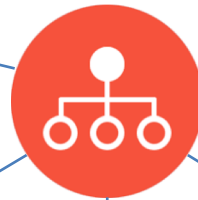
Information Authoring

Supports authoring information by different tools, which can be existing tools that have been adapted or new methodology-specific tools



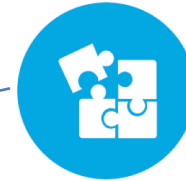
Information Representation

Supports representing information using tool-neutral ontologies that have modular design, expressive syntax & formal semantics



Information Integration

Supports defining methodology specific integrating workflows for the information fragments and proposing changes to keep them consistent



Information Configuration

Supports maintaining a baseline configuration for information (as well as change proposal configurations) and version controlling the results or running integration workflows



Information Analysis

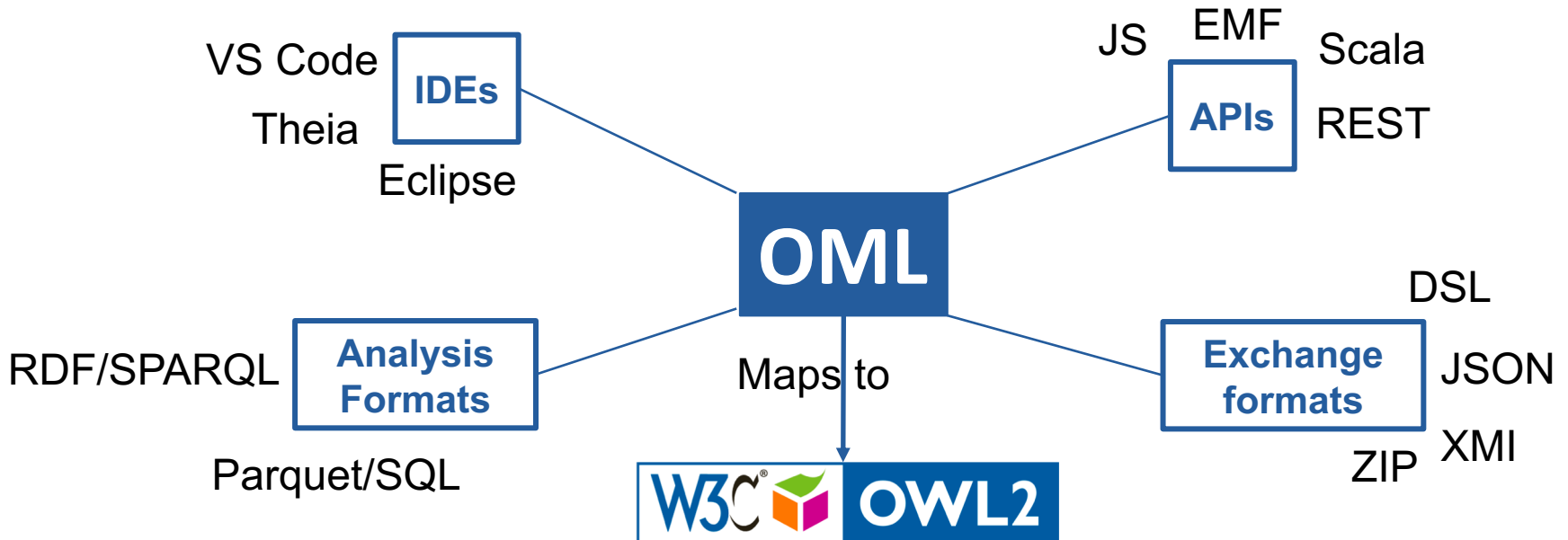
Support analyzing information with description logic reasoners and by other analysis tools using published query end points and APIs

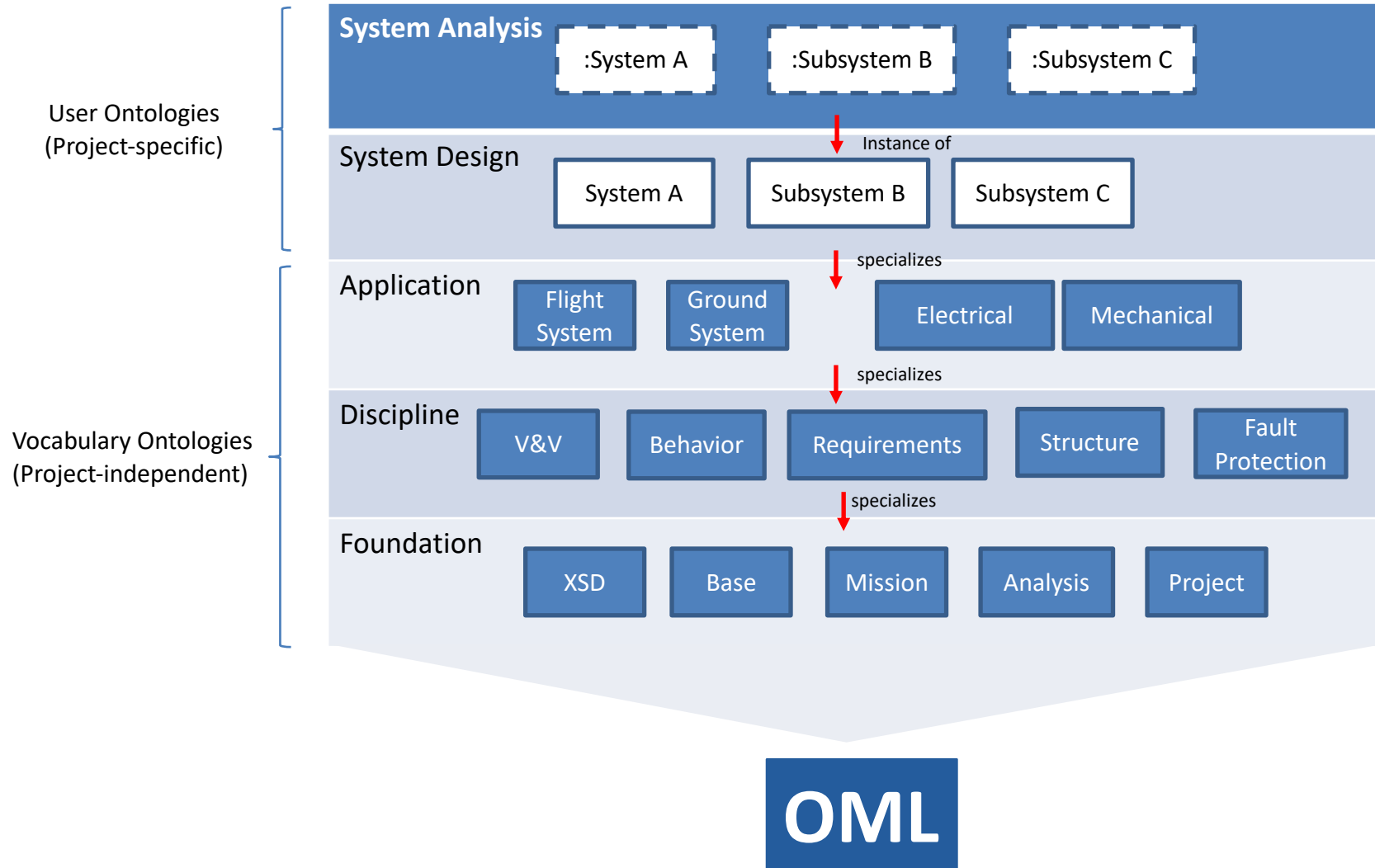


Information Reporting

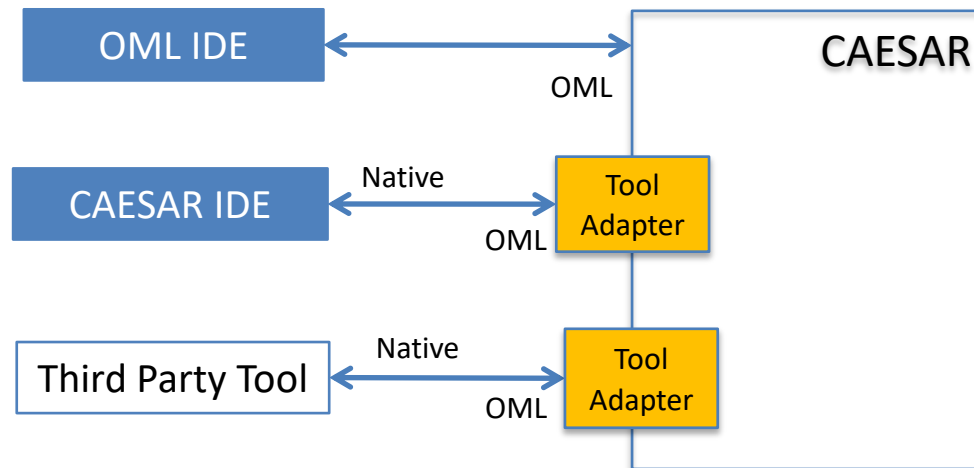
Support authoring reports on information to address different stakeholder concerns, regenerating them when the integration workflows rerun and organizing them in accessible reporting dashboards

- CAESAR represents Information as a set of ontologies expressed in the **Ontology Modeling Language (OML)**
- OML maps to the W3C standard **Web Ontology Language 2 (OWL2)**
- OML has expressive syntax, formal (**Description Logic**) semantics, and several sets of APIs, IDEs, exchange formats, and analysis formats
- OML is a tool-neutral representation that allows representing information, exchanging it, and analyzing it using a variety of tools

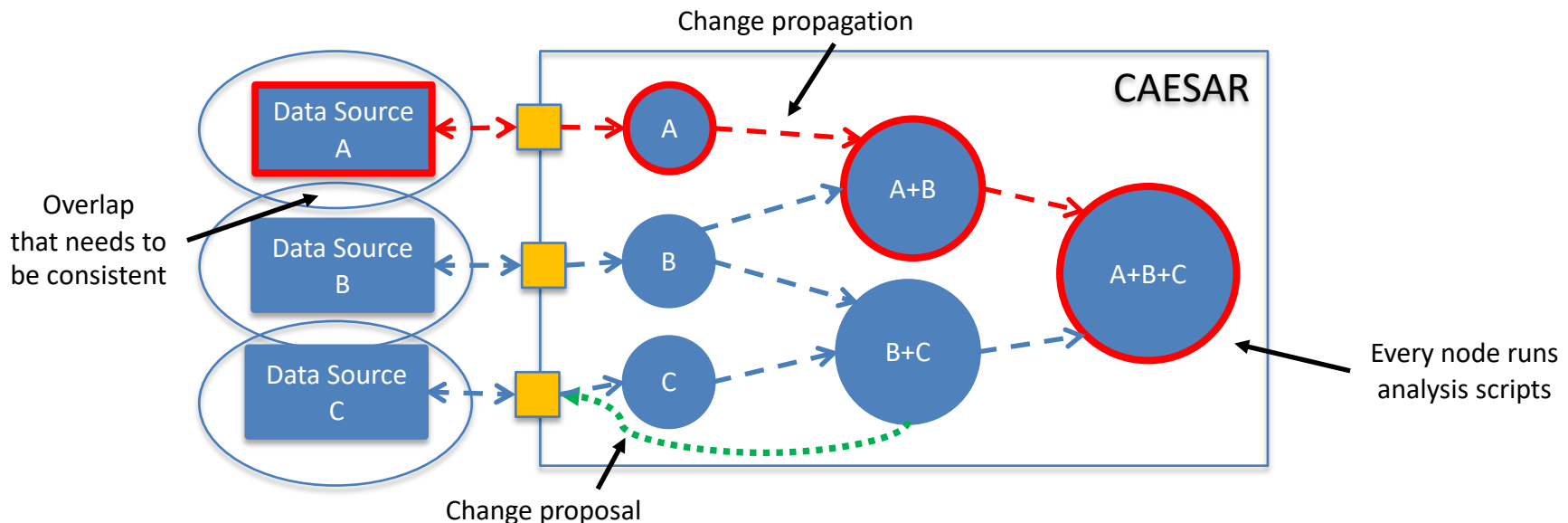




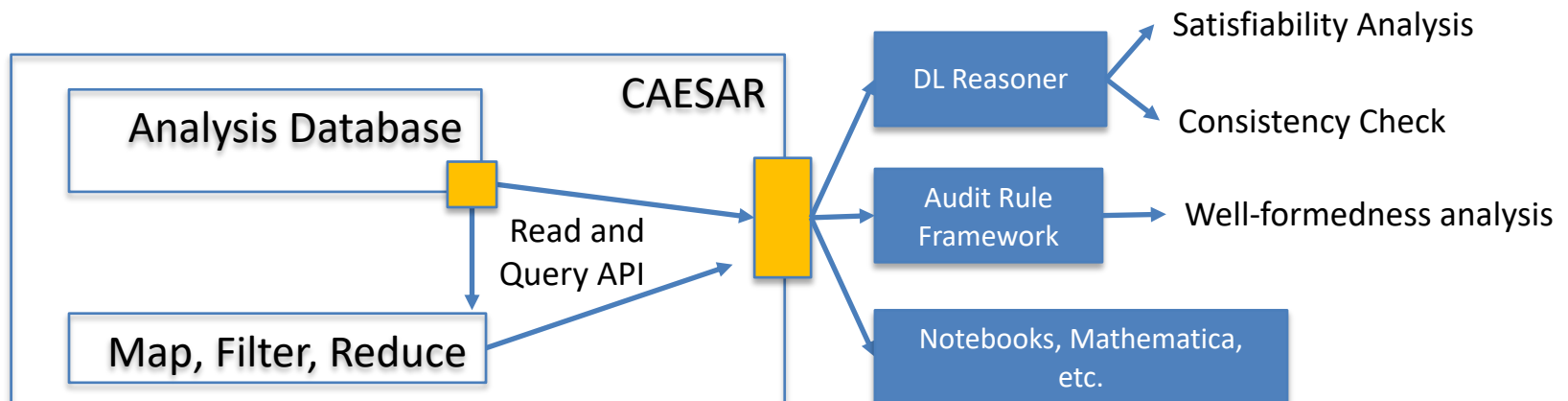
- CAESAR enables authoring ontologies directly in OML textual/graphical DSLs using one of its **OML IDEs** (Eclipse, VS Code, and Theia)
- CAESAR enables authoring ontologies using its own **CAESAR IDE**, a multi-disciplinary modeling tool which can be extended with custom viewpoints
- CAESAR also enables authoring ontologies using **third party tools** that have been adapted using a tool adapter (which supports bi-directional transformations between OML and the tool's native format)



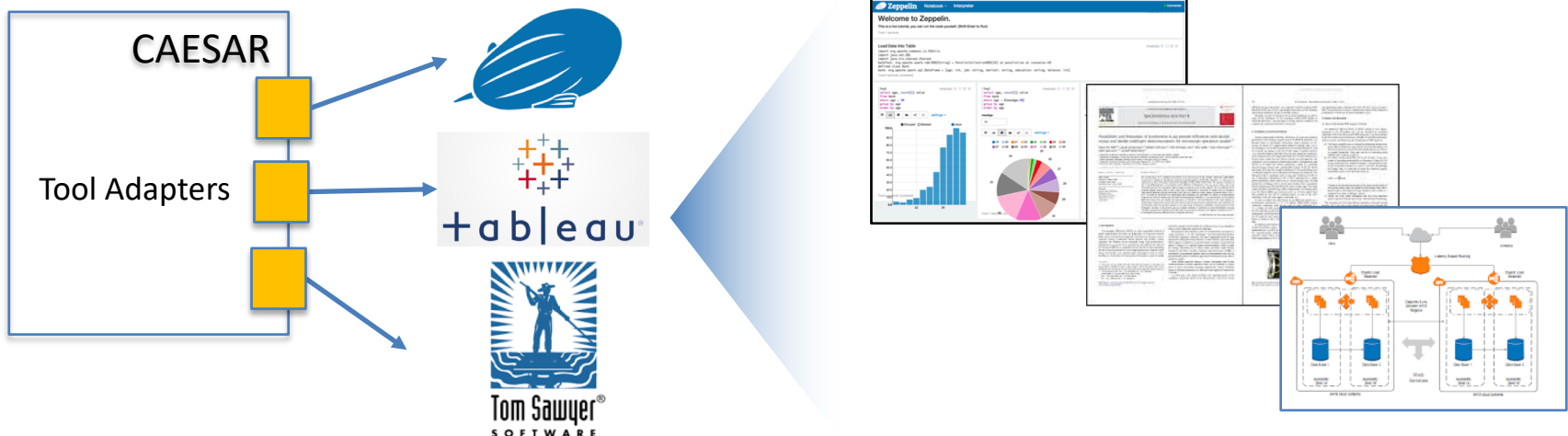
- CAESAR enables methodology-specific **incremental integration** of ontology fragments (from different tools) into a unified dataset
- The integration methodology is specified as a **workflow graph** whose nodes are configured with analysis scripts and that can be run manually or automatically based on events
- CAESAR helps maintain consistency between the integrated fragments by **proposing changes** to those fragments to restore consistency



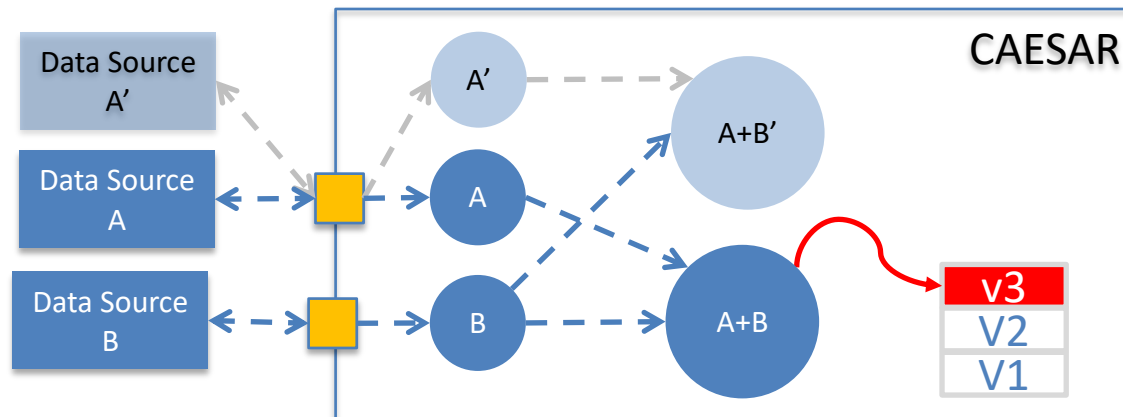
- CAESAR supports **read APIs**, a **rule language** and a **query language**, which can be used to perform powerful analysis on the dataset
- CAESAR supports some default analyses through a DL reasoner like **satisfiability analysis** (every class can be instantiated) and **consistency checking** (no contradictory axioms exist)
- CAESAR supports analysis via **third party analysis tools** by transforming the dataset to other formats expected by those tools

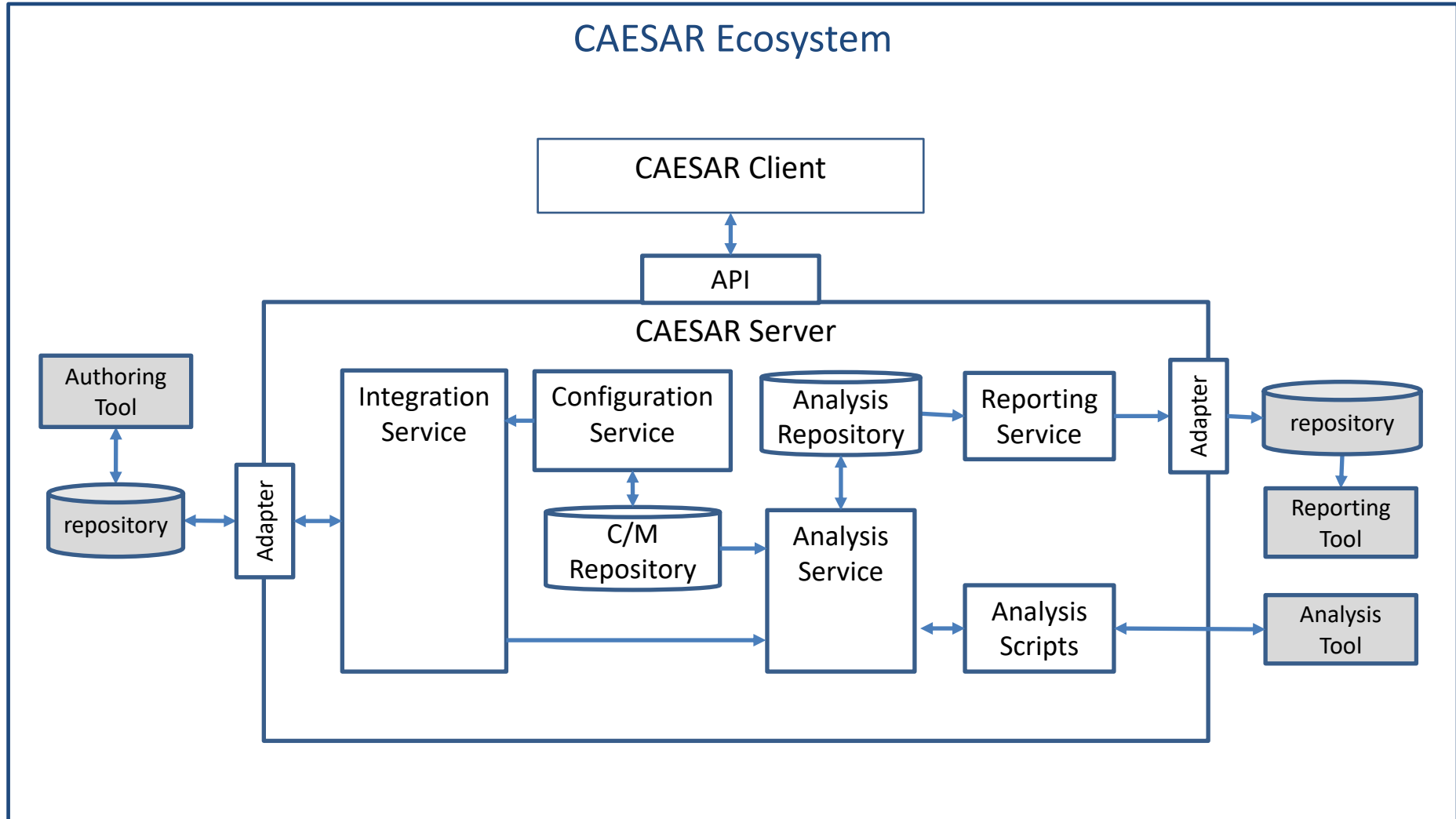


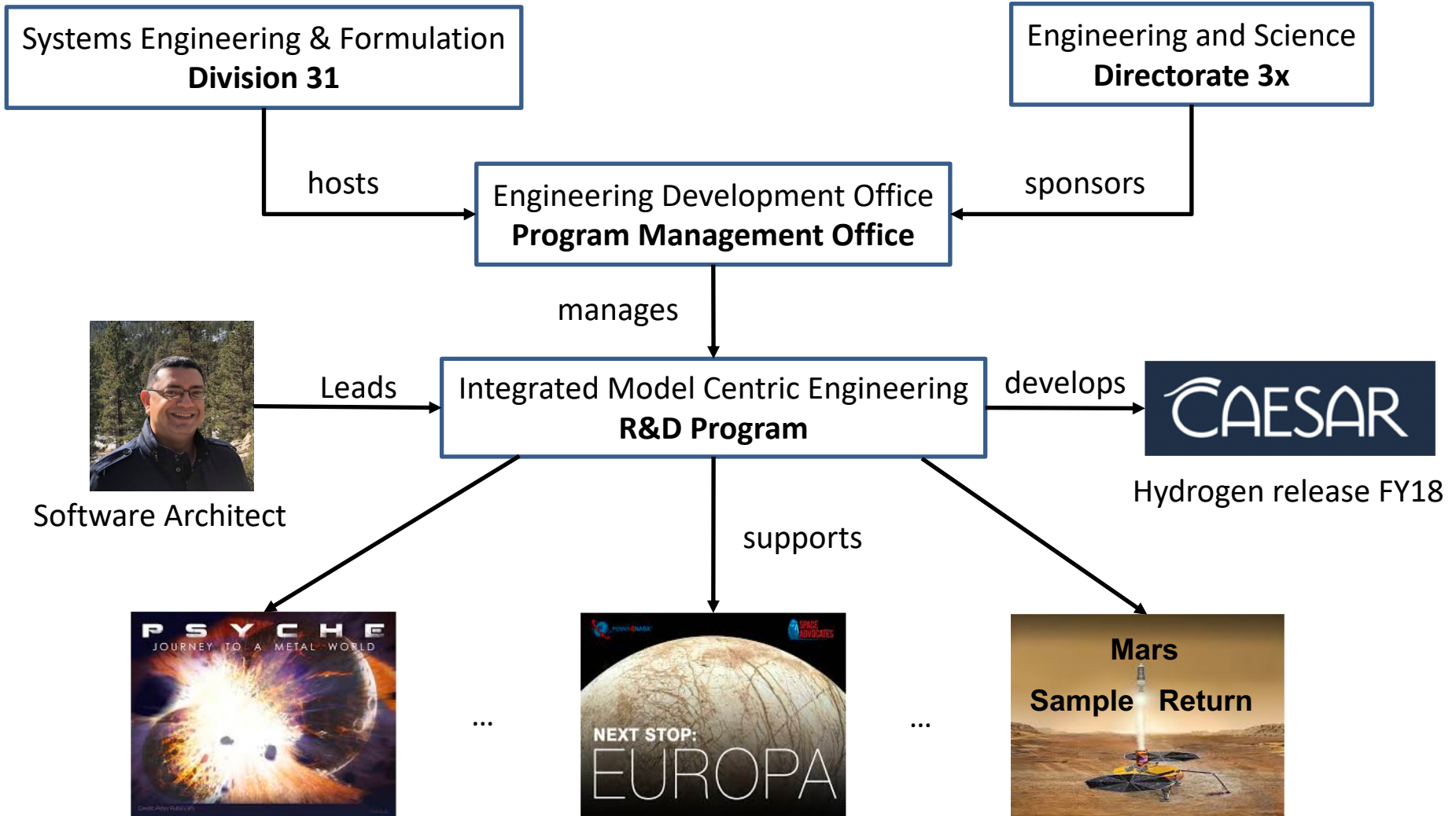
- CAESAR allows creating reports on datasets or their analysis results using a variety of **reporting tools** to address stakeholders concerns
- Reports can be automatically **produced during integration** and stored for efficient viewing later (historical reports are archived)
- Reports are **published on a project portal**, where they are organized in categories and can be searched and viewed (with access control)
- Reports can be **commented on** and/or **approved** by stakeholders

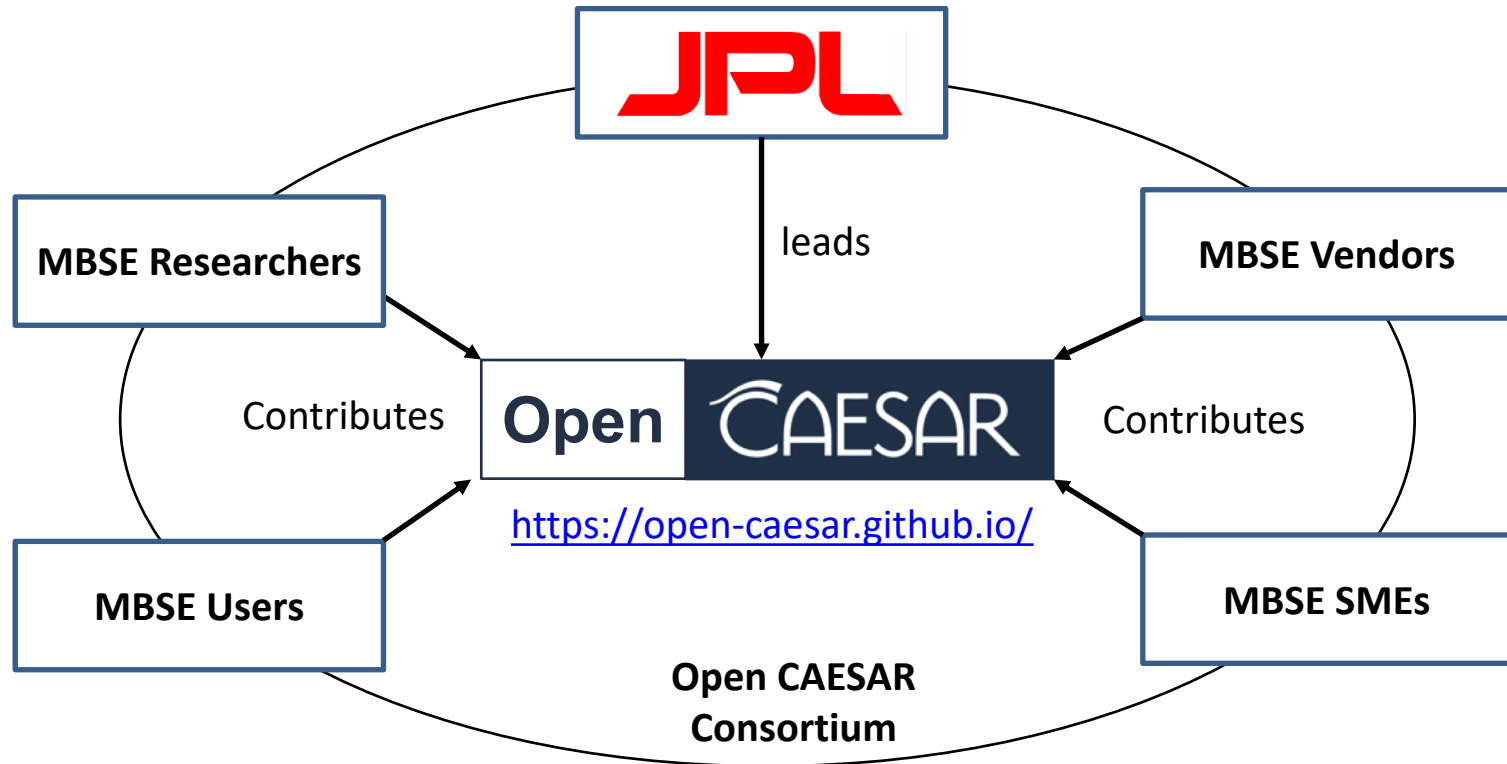


- CAESAR allows a project to have a **baseline** configuration of data sources, as well as **variant** configurations (representing change proposals)
- The project integration workflow can run on any configuration allowing the **impact of changes** to be analyzed before merging them into baseline
- The entire **history of integration** (including datasets, analysis results and reports) with full **provenance** is version controlled
- The configuration history enables **provenance audit**, **trend analysis**, as well as the possibility for **incremental analysis**





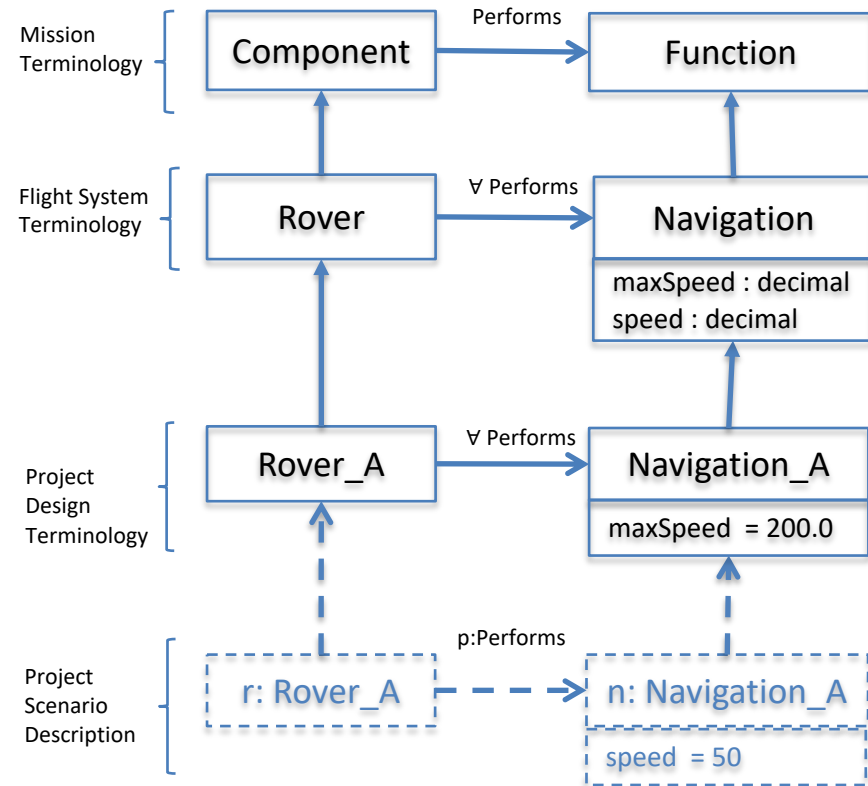




- **Information representation**
 - OML language definition and tooling
- **Information architecture**
 - Vocabularies for SE disciplines / applications
- **Information authoring**
 - Authoring tools and their adapter services
- **Information integration**
 - Infrastructure for integrating / synchronizing data sources
- **Information analysis**
 - Analysis tools and their adapter services
- **Information reporting**
 - Reporting tools and their adapter services
- **Information configuration**
 - Infrastructure for variability, change proposal, provenance
- **Example projects**
 - Case study projects to demonstrate ideas

- **Create the consortium as an entity**
 - Define consortium charter, membership rules, and legal framework
 - Create an architecture council and appoint members
 - Create a program office and appoint members
 - Adopt an open source license (e.g., Apache 2.0)
- **Manage the open source code in GitHub.com**
 - Create GitHub organization for Open CAESAR
 - Create a separate repository for each major component
 - Create a GitHub Pages site (and wiki) for each component
 - Enable CI/CD for every repository
 - Assign a development team to lead each repository
 - Define work packages as GitHub organization projects

BACKUP

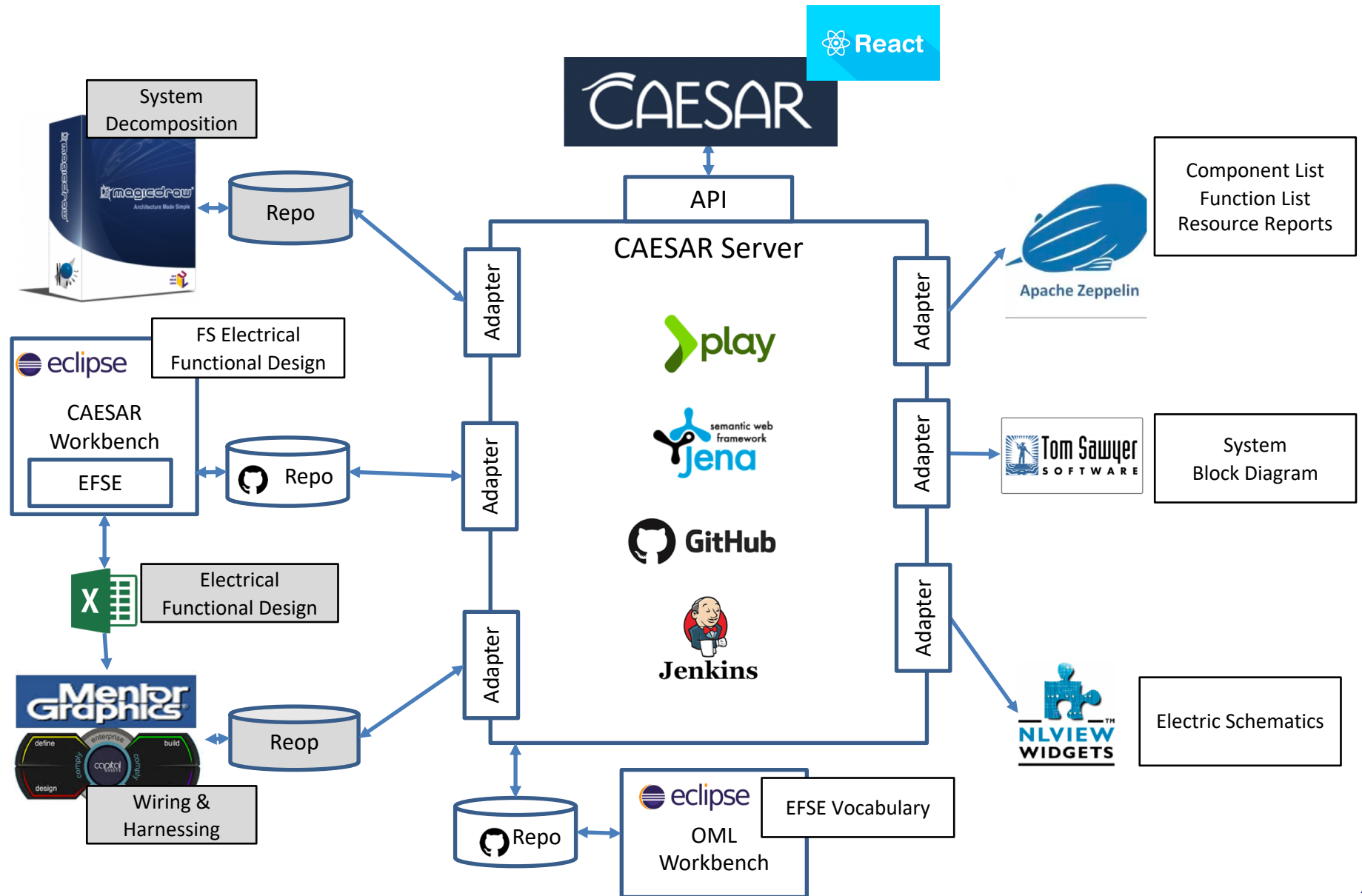


```

open terminology https://flightsystem {
  extends <https://mission>
  concept Rover {
    extends mission:Component
    restrictsAll mission:Performs to Navigation
  }
  concept Navigation {
    extends mission:Function
  }
  scalarDataProperty maxSpeed {
    domain Navigation
    range XMLSchema:decimal
  }
  scalarDataProperty speed {
    domain Navigation
    range XMLSchema:decimal
  }
}

closed terminology <https://project/design> {
  extends <https://flightsystem>
  concept Rover_A {
    extends flightsystem:Rover
    restrictsAll mission:Performs to Navigation_A
  }
  concept Navigation_A {
    extends flightsystem:Navigation
    restricts flightsystem:maxSpeed to 200.0
  }
}

closed description <https://project/scenario> {
  extends <https://project/design>
  conceptInstance r : design:Rover_A
  conceptInstance n : design:Navigation_A {
    vvalueOf flightsystem:speed is 50
  }
  relationInstance p : mission:Performs {
    from r
    to n
  }
}
  
```



The screenshot displays the OML Workbench interface with the following components:

- Project Explorer:** A tree view on the left showing the project structure. The selected path is `gov.nasa.jpl.imce.ontologies.public > converted-oml > imce.jpl.nasa.gov > foundation > base > base-embedding.oml > base.owl`.
- base.owl Editor:** The central pane shows the OWL ontology code. It includes imports, aspects for `AggregatedElement`, `ContainedElement`, `Container`, and `IdentifiedElement`, and reified relationships for `Aggregates` and `Contains`.
- Terminology Diagram:** A diagram on the right visualizing the ontology structure. It shows classes `base:AggregatedElement`, `base:ContainedElement`, `base:IdentifiedElement`, and `base:Package` with their relationships and properties.
- Outline:** A pane at the bottom left showing the hierarchy of the selected ontology, including `Extent`, `OpenWorldDefinitions`, and `TerminologyGraph(base)`.

A large blue box with the text "OML Workbench" is overlaid on the right side of the interface.

runtime-caesar.product - platform:/resource/gov.nasa.jpl.europa.efse/functionlist/europa_functionlist.aird/Function List Table - CAESAR

Quick Access

Model Explorer

type filter text

- gov.nasa.jpl.europa.efse [europa.efse issue-31]
 - Project Dependencies
 - flightsystem
 - europa_flightsystem.aird
 - europa_flightsystem.refdes
 - europa_flightsystem.xlsx
 - functionlist
 - europa_functionlist.aird
 - europa_functionlist.cefl
 - europa_functionlist.xlsm
 - libraries
 - AssemblyTypes.library
 - ECTypes.library
 - EuropaAssemblyTypes.library
 - FunctionTypes.library
 - representations.aird

*Function List Table

Function Number	Function Name	Subsystem 1	Assembly 1	Flow	Subsystem
Function List					
Subsystem Pair 2001-2004 (AMSTRU-PWR)					
Subsystem Pair 2001-2006 (AMSTRU-CDH)					
001-006-001	TLM VSA MXP EBOX IF TEMP TO REU-A	2001	TSMSIF-A	>	2006
001-006-002	TLM VSA MXP EBOX IF TEMP TO REU-B	2001	TSMSIF-B	>	2006
001-006-003	TLM VSA MXP RDMSA IF TEMP TO REU-A	2001	TSRMIF-A	>	2006
001-006-004	TLM VSA MXP RDMSA IF TEMP TO REU-B	2001	TSRMIF-B	>	2006
	REU-A	2001	TSB1F-A	>	2006
	REU-B	2001	TSB1F-B	>	2006
	REU-A	2001	TSB2F-A	>	2006
	REU-B	2001	TSB2F-B	>	2006
	-A	2001	TSVMZ-1A	>	2006
	-B	2001	TSVMZ-1B	>	2006
	A	2001	TSVMZ-2A	>	2006
	B	2001	TSVMZ-2B	>	2006
	A	2001	TSVMZ-3A	>	2006
	B	2001	TSVMZ-3B	>	2006
		2001	TSVMZ-4A	>	2006
		2001	TSVMZ-4B	>	2006
	J-A	2001	TSVPX-A	>	2006
	J-B	2001	TSVPX-B	>	2006

CAESAR hydrogen
October 2018

Outline

An outline is not available.

Properties

Function

General

Annotations

Properties

Function Ends

Name: TLM VSA MXP EBOX TEMP TO REU-A

Specializes: Abstract Function PRT2Measurement

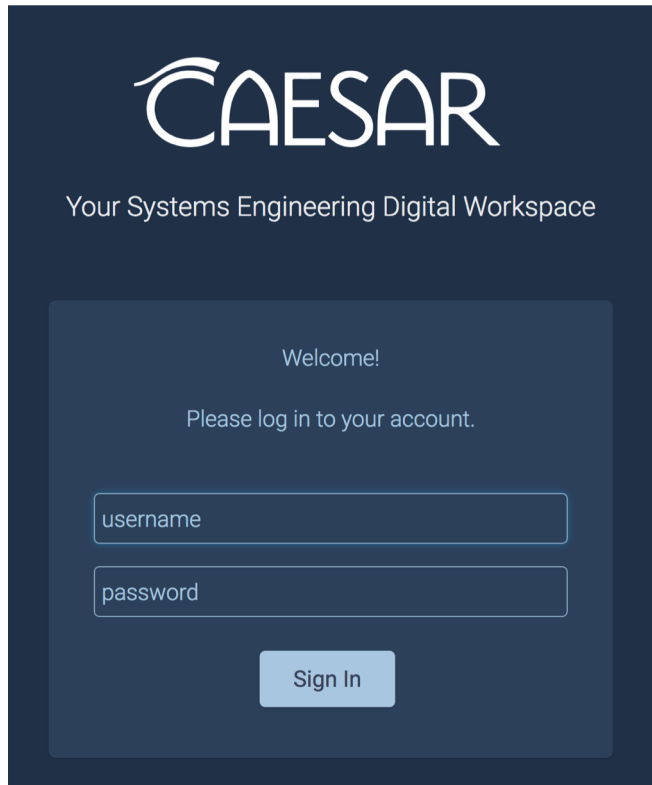
Is Deprecated

Subsystem Pair: SubsystemPair: Subsystem Pair 2001-2006 (AMSTRU-CDH)

Serial Number: 9

Flow: Undefined From1to2 From2to1 Bidirectional

Assemblies1: Assembly 2001TSVMZ-1A



The login screen features the CAESAR logo at the top, followed by the tagline 'Your Systems Engineering Digital Workspace'. Below this, a 'Welcome!' message and a prompt 'Please log in to your account.' are displayed. There are two input fields for 'username' and 'password', and a 'Sign In' button at the bottom.

CAESAR

Your Systems Engineering Digital Workspace

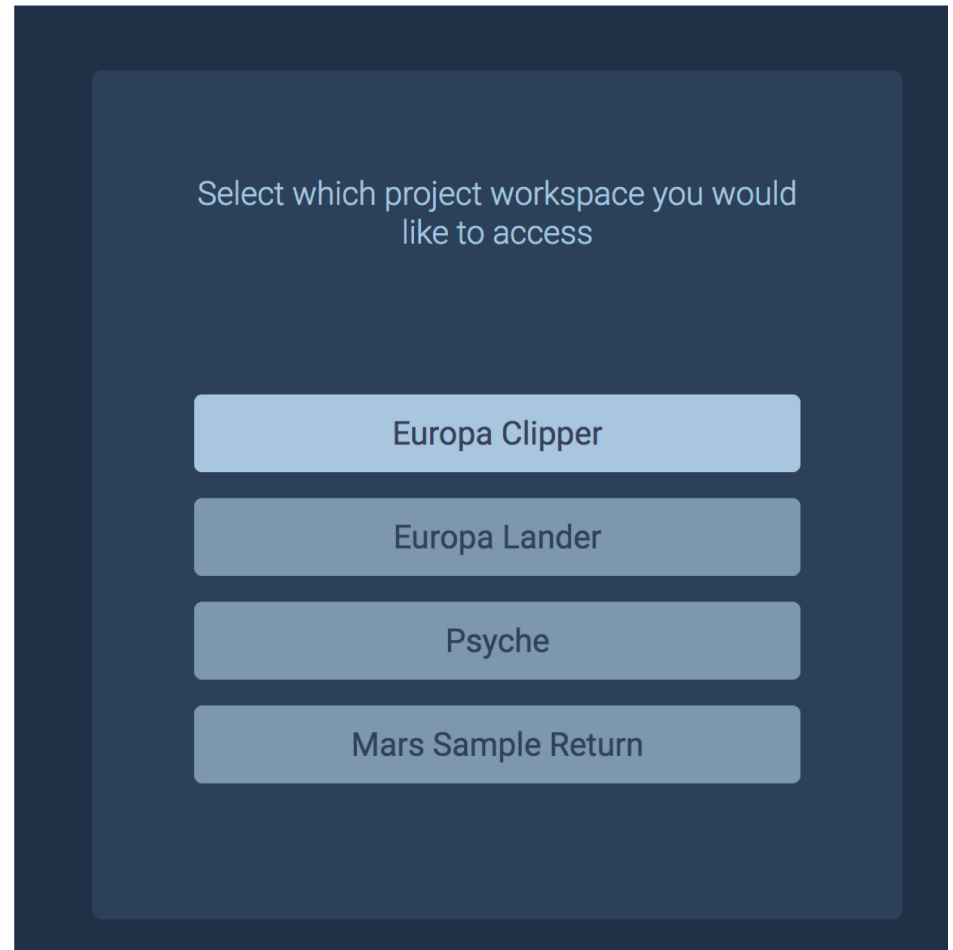
Welcome!

Please log in to your account.

username

password

Sign In



The project selection screen displays the instruction 'Select which project workspace you would like to access'. Below this, there are four buttons representing different project workspaces: 'Europa Clipper', 'Europa Lander', 'Psyche', and 'Mars Sample Return'.

Select which project workspace you would like to access

Europa Clipper

Europa Lander

Psyche

Mars Sample Return

Reports

Select a report to view

Search by Report Name

Technical Resource Reports
This collection of reports include multiple types of Technical Resource Reports

Function List
Function List data from EFSE

Reference Designator List
Reference Designator data from EFSE

Component List

EUROPA CLIPPER REPORTING PERSPECTIVE

< Function List

Function List data from EFSE

Search by Report Name

Branch: Baseline Latest

Version: This is a dynamic endpoint for latest data

Timepoint: 2018-08-27T15:00:01

View Report

IMCE's Zeppelin Notebook Job

templates/EFSE Reports/Function List/

functi dataset

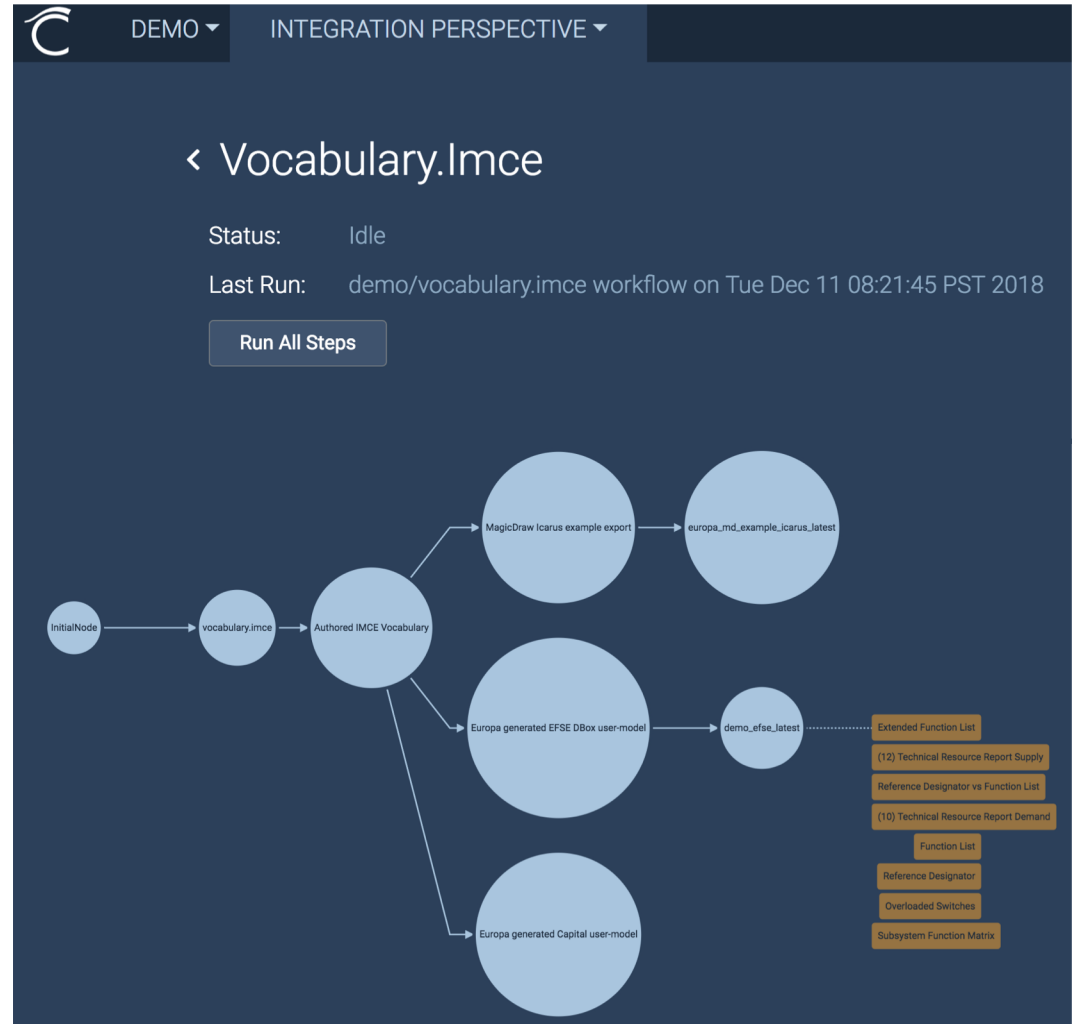
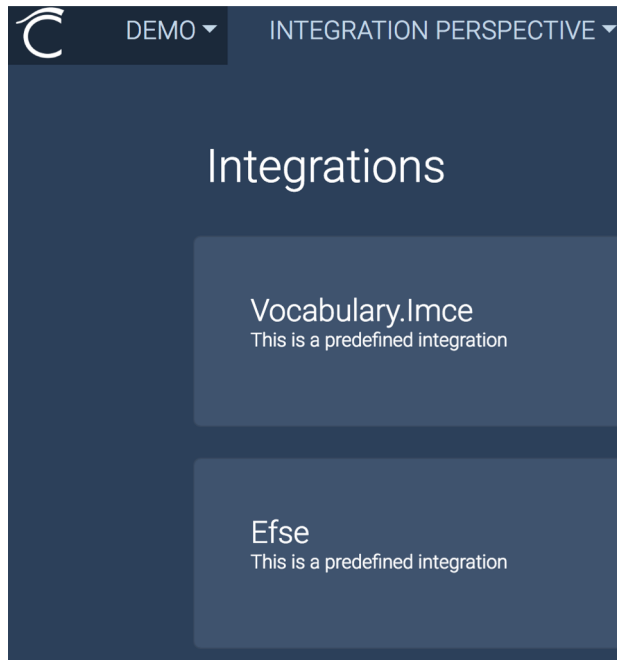
https://imce-caesar-repository.git.nasa.gov/ c8278c82cc16149d9f9851e8a0c30f195374ef

warnings: there were two deprecation warnings: re-run with --deprecation for details
functi: Object = https://imce-caesar-repository-381.nasa.gov/
dataset: Object = c825c8c3c6149e9ff6851e8a0c30f195374ef

Function List Report from EFSE

The table shows all the Function List data from EFSE.
To download the data in .csv format click the right corner of the table and select download as .csv Note: Not all data types will support download to excel

FUNCTION_NUMBER	FUNCTION_NAME	SUBSYSTEM_1	ASSEMBLY_1	FLOW	ASSEMBLY_2	SUBSYSTEM_2
001-004-001	BATT PWR TO BATTARMP PTH	2001	2001BATTARMP	<	2004BATT-1, 2004BATT-2, 2004BATT-3	2004
001-004-002	BATT PWR BATTARMP PTH TO ABIS	2001	2001BATTARMP	>	2004ABIS	2004
001-004-003	PROP PWR A ABIS TO DPVENAP PTH	2001	2001PDRARMP-A	<	2004ABIS	2004
001-004-004	PROP PWR B ABIS TO DPVENAP PTH	2001	2001PDRARMP-B	<	2004ABIS	2004
001-004-005	DPY PWR A ABIS TO DPVENAP PTH	2001	2001PDRARMP-A	<	2004ABIS	2004



2AHIF-B

SUBSYSTEM	TOTALFUNCTION	2A	4A	5A
AMSTRU	79	6	0	0
CDH	958	0	0	4
EIS	94	13	0	1
ETHEMIS	21	2	0	1
GNC	210	4	0	4
ICEMAG	46	7	0	0
IMB	0	0	0	0
MASPEX	29	4	0	2
MISE	31	6	0	1
PIMS	29	4	0	0
PME	488	2	0	0
PMHARN	4	0	0	0
PMSTRU	614	0	0	0
PROP	340	0	0	0

Function List Report

The table shows all the Function List data.
To download the data in .csv format click

FUNCTION_NUMBER	FUNCTION_NAME	FUNCTION_DESCRIPTION	FUNCTION_NUMBER	FUNCTION_NAME	FUNCTION_DESCRIPTION
004-004-001	TLM BATT-1 V	PBC	004-004-002	TLM BATT-2 V	PBC
004-004-003	TLM BATT-3 V	PBC	004-004-004	TLM BATT-1 I	PBC PTH
004-004-005	TLM BATT-2 I	PBC PTH	004-004-006	TLM BATT-3 INTEMP	PBC PTH
004-004-007	TLM BATT-1 INTEMP	PBC	004-004-008	TLM BATT-2 INTEMP	PBC
004-004-009	TLM BATT-3 INTEMP	PBC	004-004-010	UMB CMD BATT SW 1	

The image illustrates the development environment for the Open CAESAR OML Language Server, showing four different ways to interact with the code:

- Gitpod:** A web-based IDE accessible via a browser, showing the OML code and a visual graph view of the model.
- Eclipse:** A traditional IDE showing the OML code and a properties outline.
- VS Code:** A modern IDE showing the OML code and a file explorer.
- GitHub:** The source code repository for the project.

A green arrow indicates the flow from the Gitpod web editor to the VS Code IDE.

- ***Basic tooling:***

- OML Abstract syntax, Java API, XML interchange format (EMF)
- OML Textual syntax / language server (Xtext)
- OML Graphical syntax (Sprotty)
- OML Semantics by mapping to OWL (Jena)

- ***Other APIs:***

- OML Scala functional API
- OML Java script API
- OML REST API

- ***Other interchange formats:***

- OML JSON format
- OML Parquet format
- OML Zip format

- ***Foundation vocabularies:***
 - Vocabulary for architecture description
 - Vocabulary for architecture integration
 - Vocabulary for diagram representation
- ***SE discipline vocabularies:***
 - Vocabulary for requirements
 - Vocabulary for use cases
 - Vocabulary for system structure
 - Vocabulary for system behavior
 - Vocabulary for V&V
 - Vocabulary for I&T
- ***SE application vocabularies:***
 - Vocabularies for different application domains (aerospace, automotive, etc.)

- ***OML IDEs:***

- OML IDE support in Eclipse (textual editor, graphical viewer, EMF frameworks)
- OML IDE support in Theia (textual editor, graphical viewer, Gitpod integration)
- OML IDE support in VS Code (textual editor)

- ***EMF-based tools:***

- New EMF-based tool for existing OML vocabulary
- New OML vocabulary for an existing EMF-based tool
- Generic OML adapter CLI based on OML-EMF vocabulary mapping

- ***UML-based tools:***

- New UML-based tool for existing OML vocabulary
- New OML vocabulary for an existing UML-based tool
- Generic OML adapter CLI based on OML-UML vocabulary mapping

- ***Other tools:***

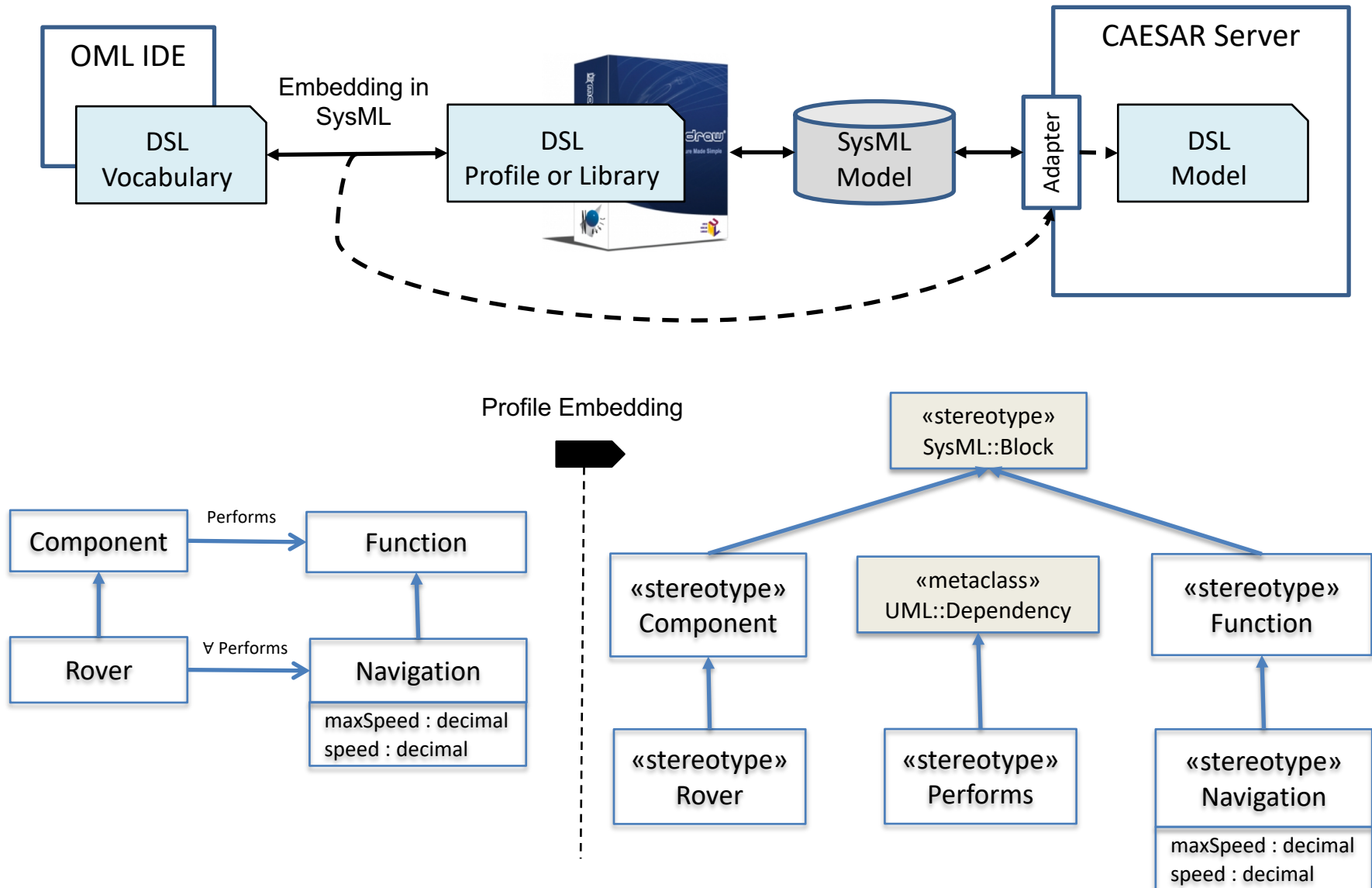
- Framework for an OML adapter CLI
- Template generator for an OML adapter CLI

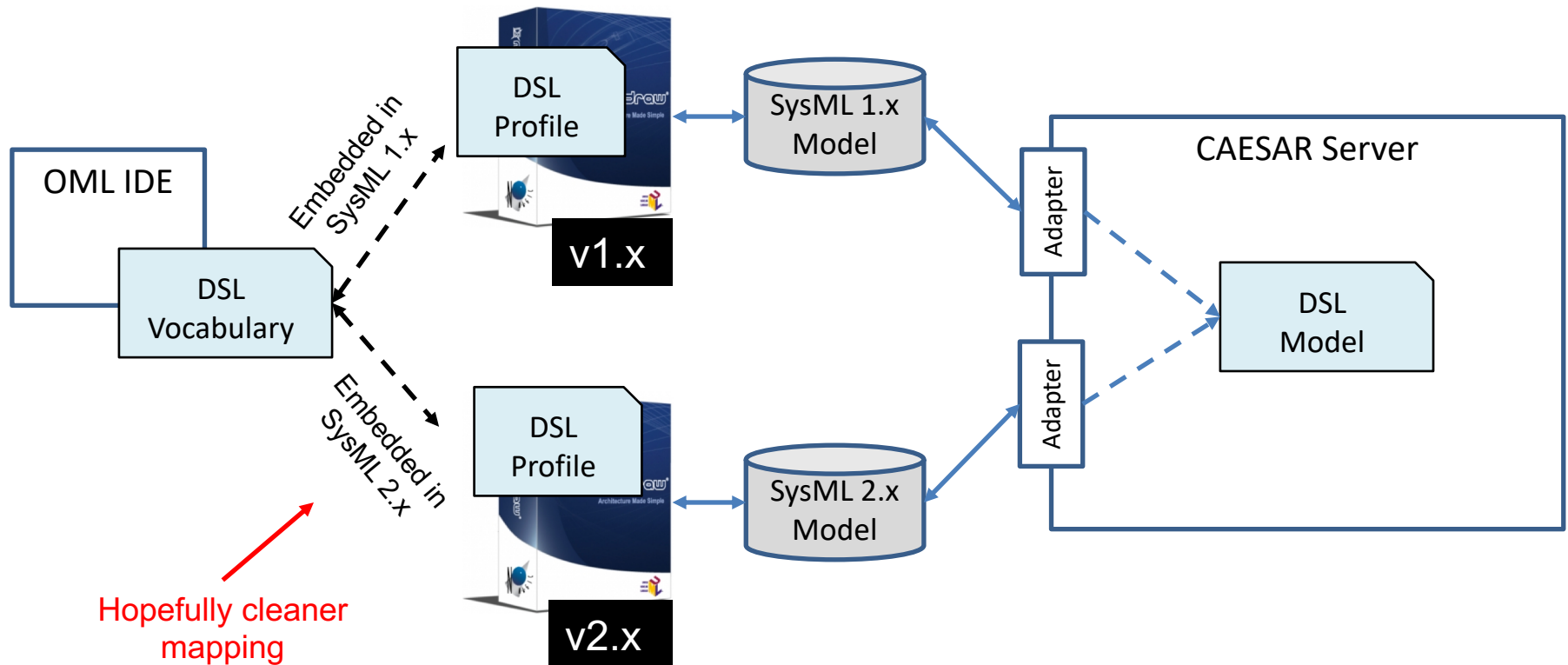
- ***Continuous integration and delivery:***
 - Vocabulary for describing a model integration workflow
 - Service to run a model integration workflow
 - Service to publish the results of a model integration flow
- ***Proposing change deltas to integrated SE tools:***
 - Way to represent change delta in OML
 - Framework to propose a change delta in OML
 - Framework to apply the change delta in a given tool

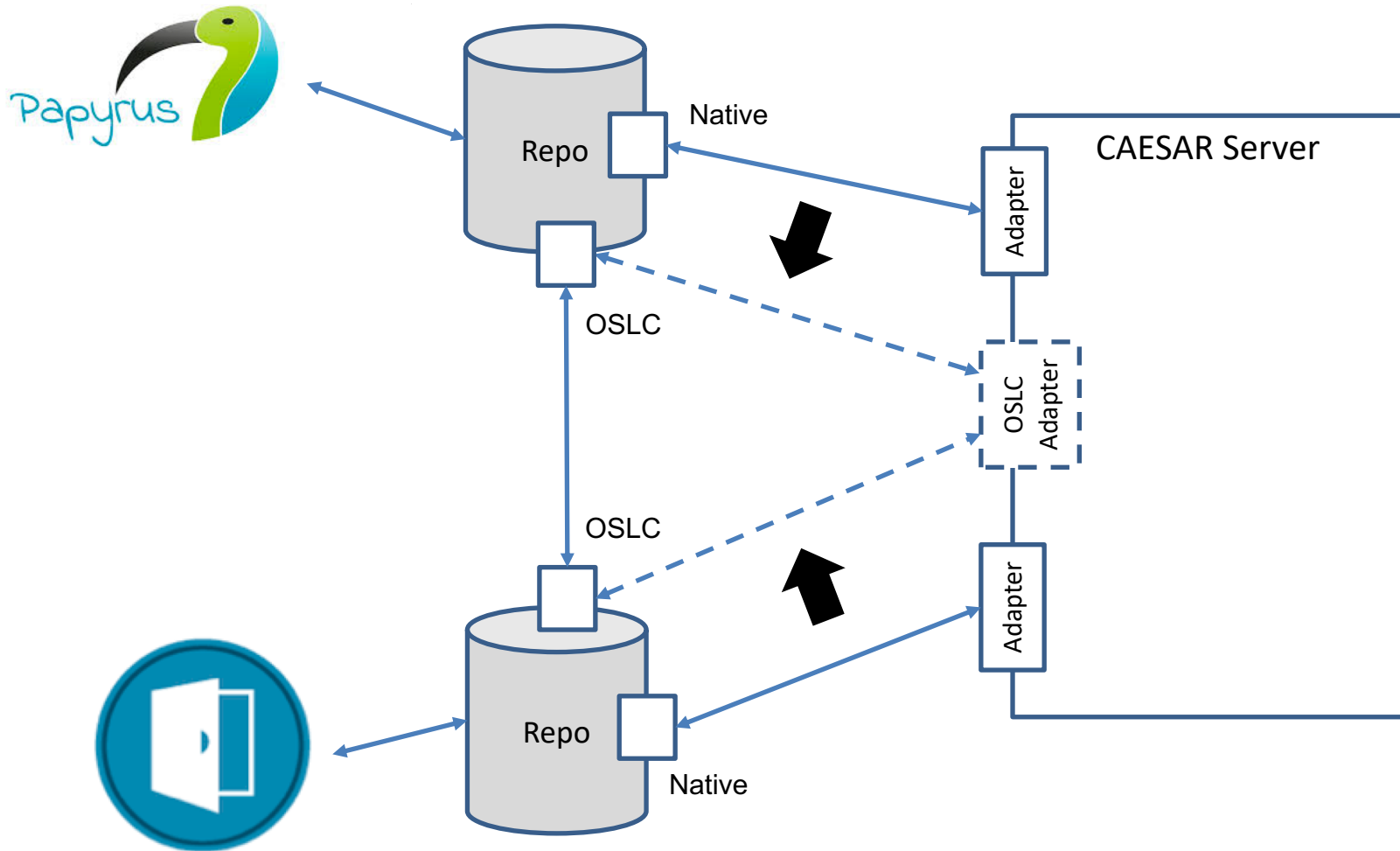
- ***Analysis on OML representation directly:***
 - Programmatic query API for OML
- ***Analysis using a triple store:***
 - Way to publish OML ontologies into a triple store
 - SPARQL endpoint for OML
 - Way to reason on OML with a DL reasoner
 - Higher level query language for OML
- ***Analysis using relational database:***
 - Way to publish OML ontologies into a relational database
 - SQL endpoint for OML
- ***Analysis using other analysis tools:***
 - Framework to analyze OML with other analysis tools
 - Framework for descriptive, predictive, prescriptive analytics

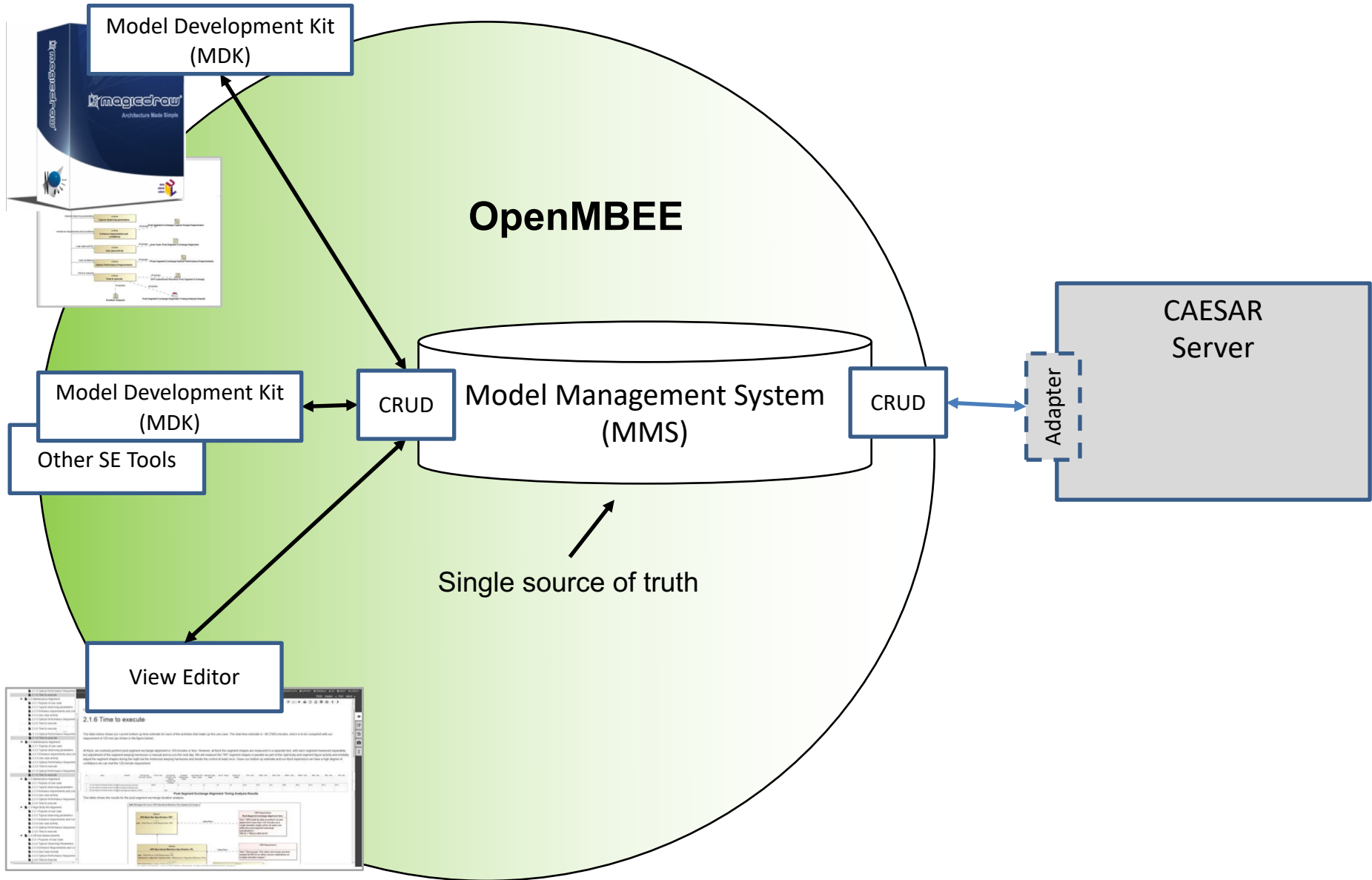
- ***Ontology visualization on the web:***
 - Web-based widgets for visualizing OML ontologies
- ***Ontology reporting tools:***
 - Interfaces for OML ontologies in various reporting tools
 - Way for reports to display change deltas
 - Way to report on trends in the evolution history
- ***Reporting dashboards:***
 - Way to organize reports to make it easy to browse/search
 - Way to review and comment OML ontologies
 - Way to approve changes to OML ontologies
 - Way to subscribe to changes in reports
- **Interactive Reporting tools**
 - Tool to browse the contents of the OML ontologies
 - Tool to assess impact of change to OML ontologies
 - Tool to run (elastic) text search on OML ontologies

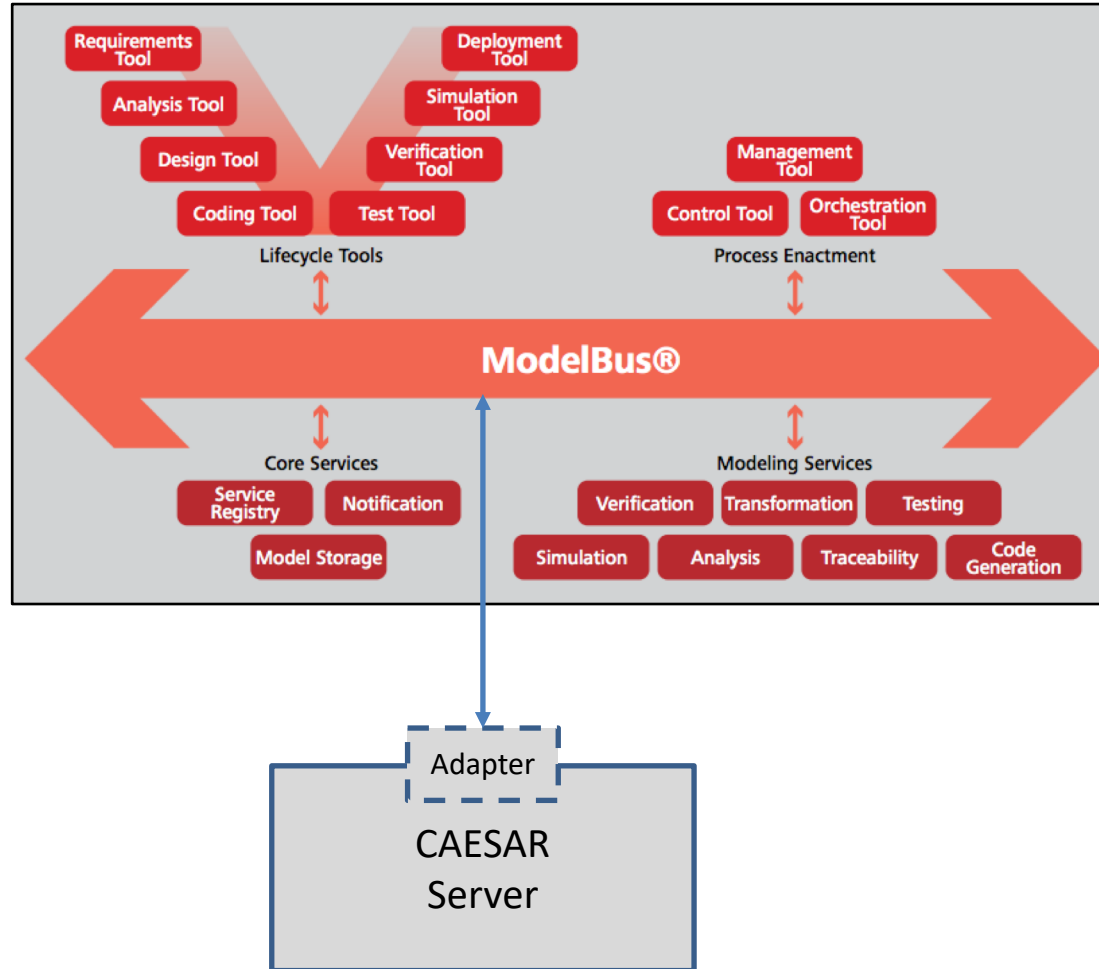
- ***Configuration management of ontologies:***
 - Way to manage OML ontologies in a SCM repository
 - Way to compare and merge ontologies in different repository configurations
 - Way to propose changes and doing pull requests using configurations

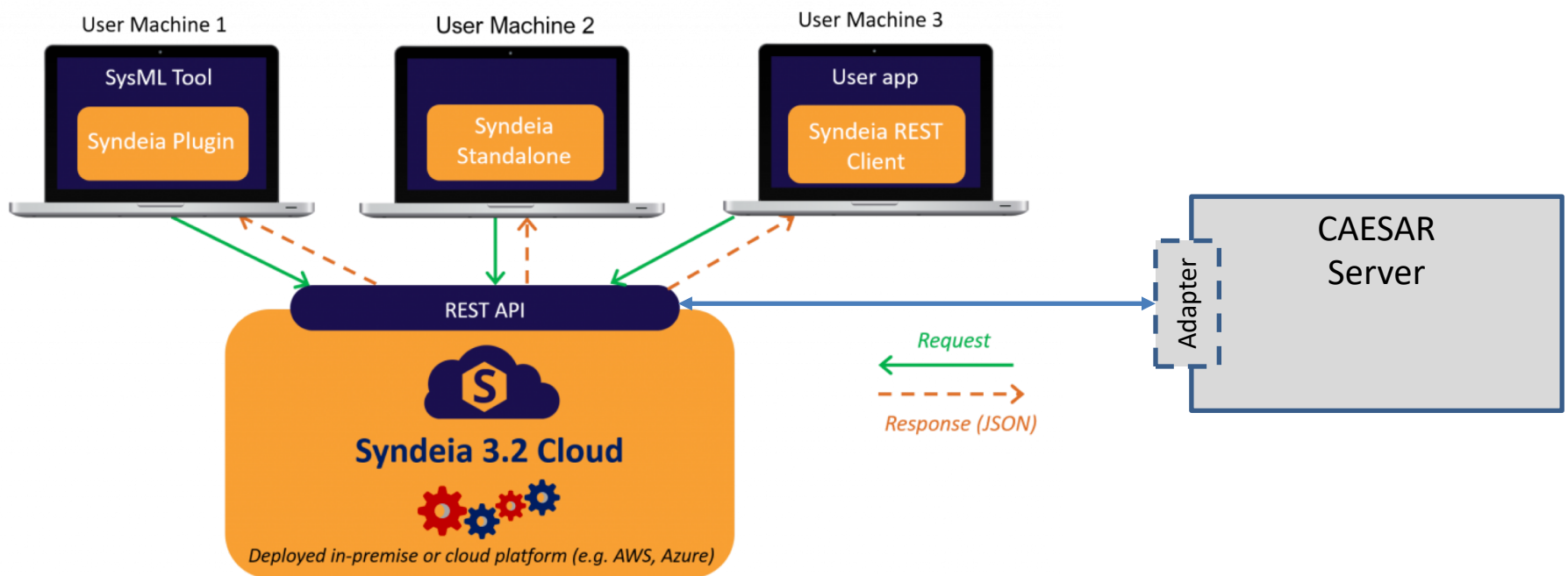








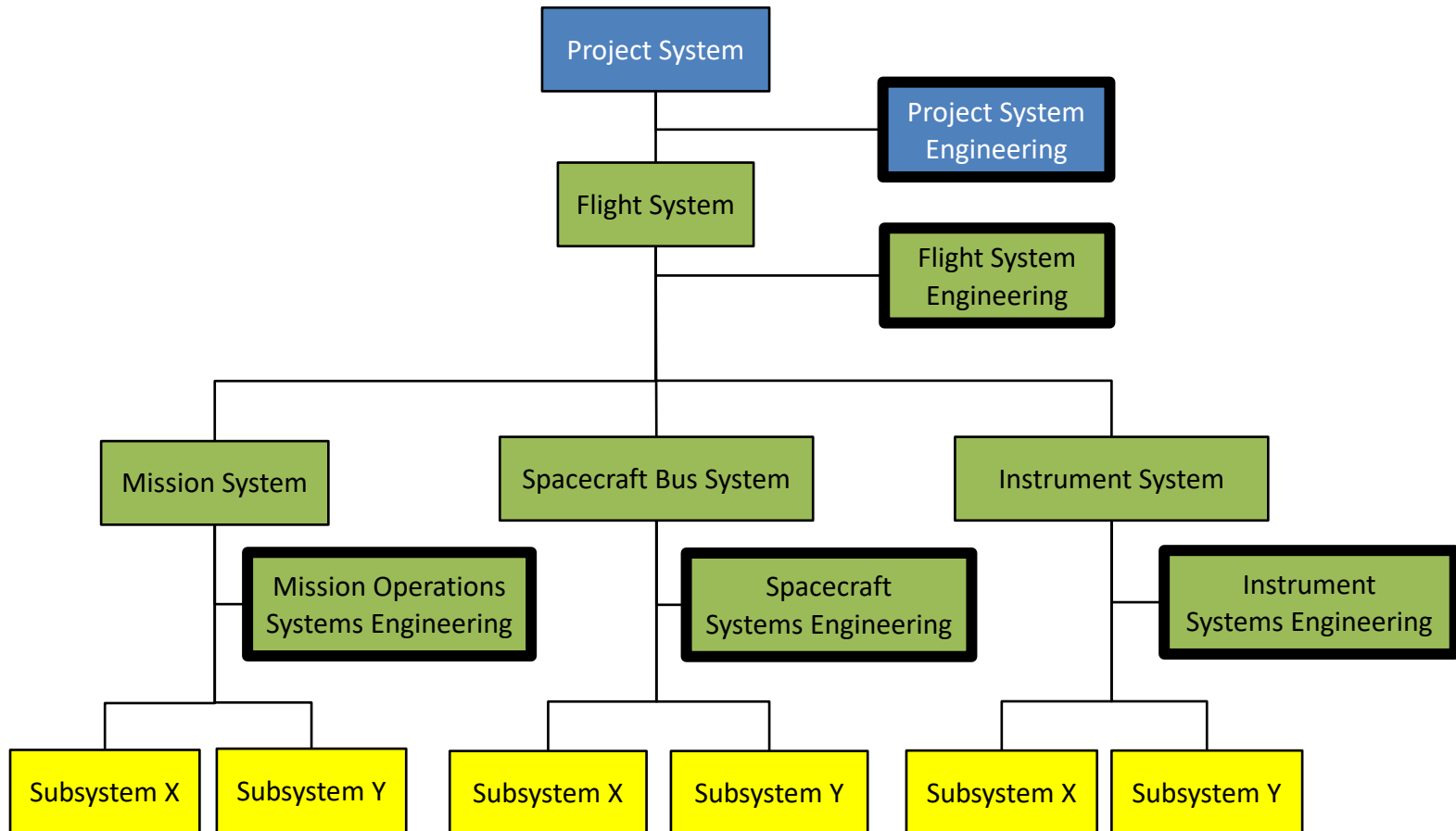




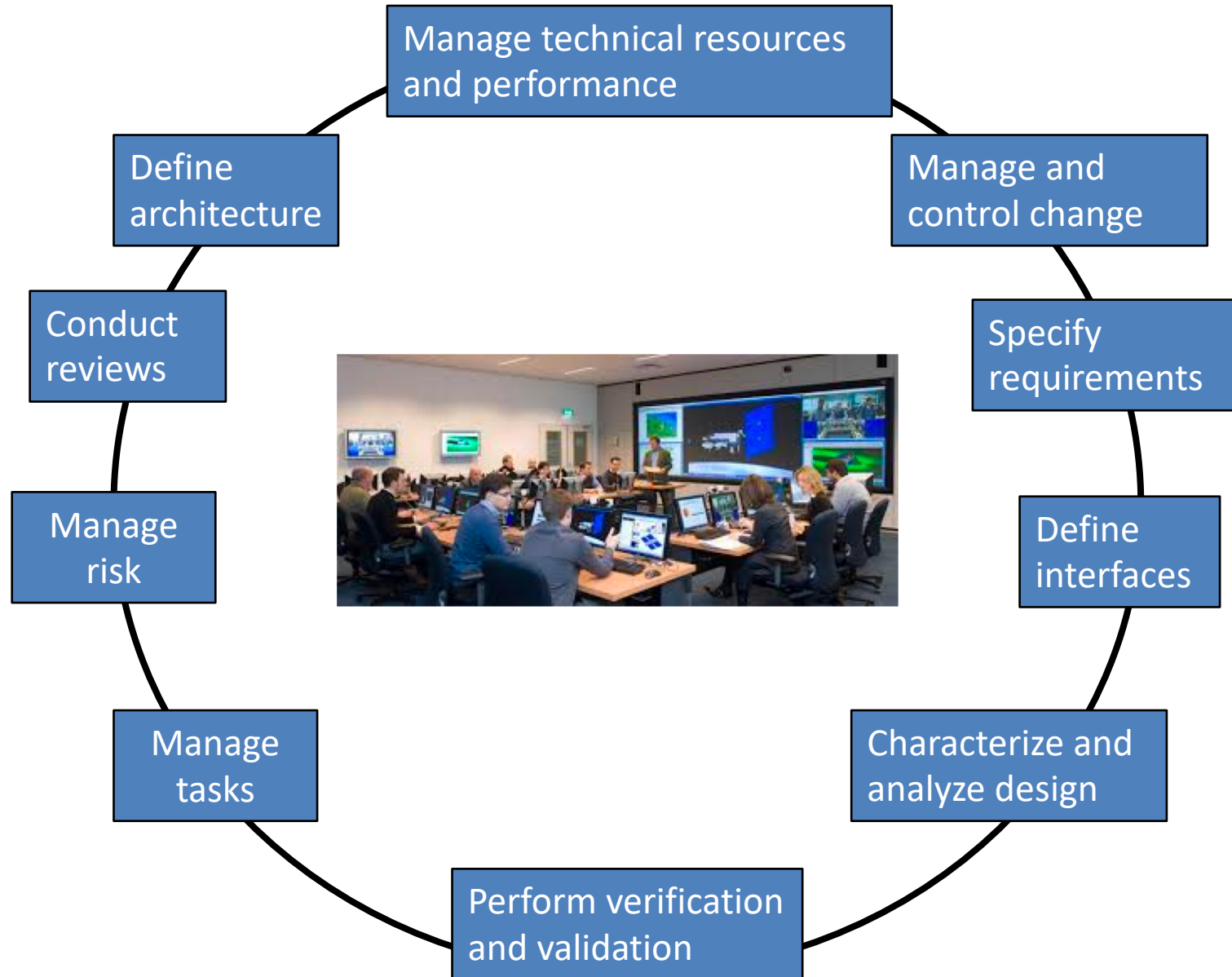
Program:
Level 1

Project:
Level 2

System:
Level 3



Subsystem:
Level 4



Electrical Systems & Signals

- Compositional System Design Description Capture
- System Block Diagram
- Function Lists, Net Lists
- Electrical Flight Systems Engineering
 - Circuit Data Sheets

Functional and Behavioral Design

- Scenario Definition, Planning and Analysis
- Fault Protection Monitors and Responses
 - Functional Description Document Generation
 - Specification of FSW behavioral design
- Power scenario analysis
- Data scenario analysis

Physical & Mechanical Composition

- Compositional System Design Description Capture
 - Master Ref. Des. list reconciled with drawing tree
- Mass Management Process / MEL
 - Mass Properties Management
 - Mapping Assembly aggregations vs. Subsystem Aggregations